Network Simulator:
NS-3

Kameswari Chebrolu

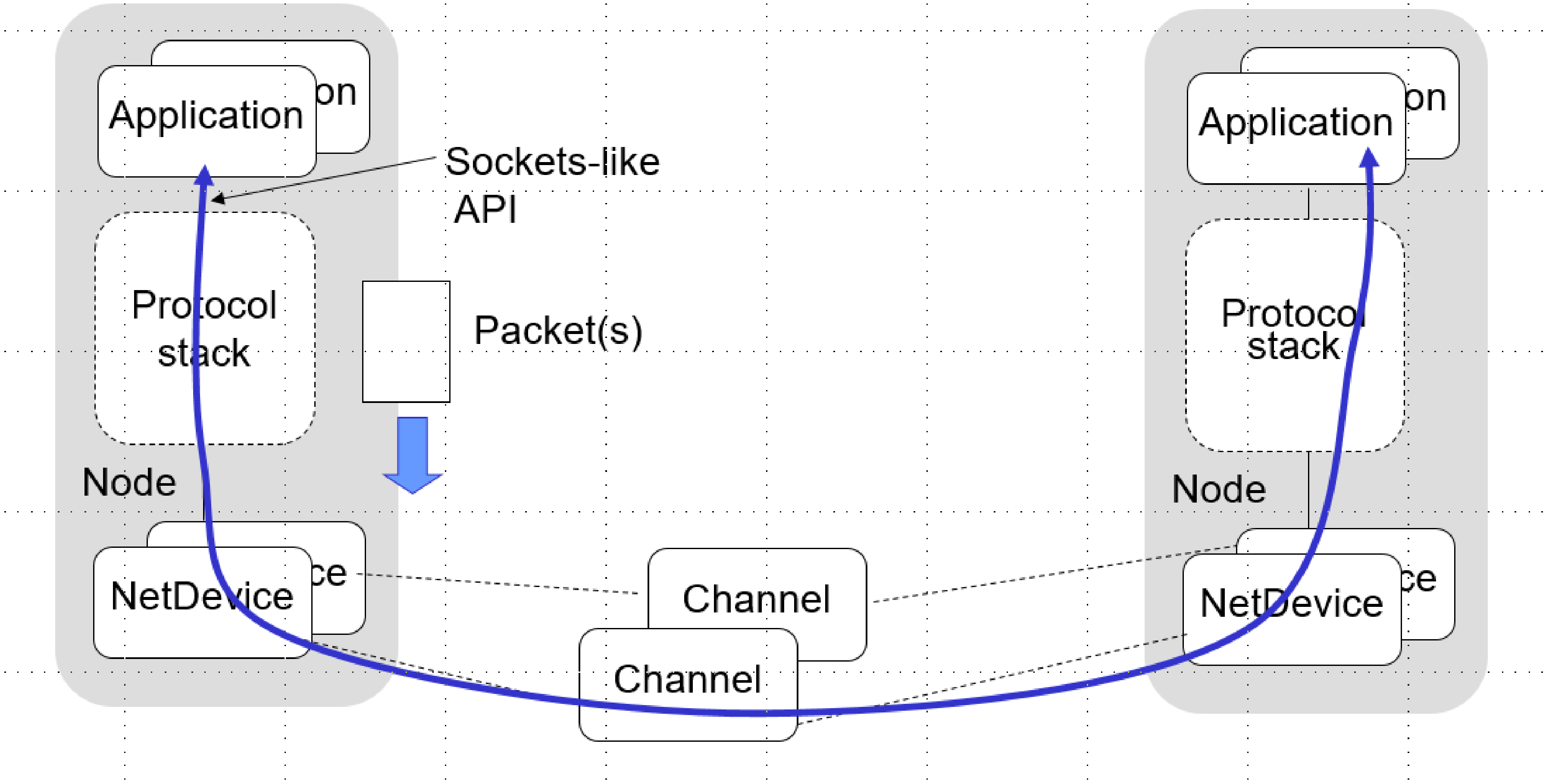
Reference: ***https://www.nsnam.org/tutorials/simutools08/ns-3-tutorial-slides.ppt***
https://www.nsnam.org/docs/release/3.2/tutorial/tutorial_18.html

Motivation for Simulations

- Cheap -- does not require costly equipment
- Complex scenarios can be easily tested
- Results can be quickly obtained – more ideas can be tested in a smaller timeframe
- The real thing isn't yet available
- Controlled experimental conditions
  - Repeatability helps aid debugging
- Disadvantages: Real systems too complex to model

Features of NS-3

- Protocols: Propagation Models, 802.11 MAC , OSLR, TCP, UDP, OnOff Applicaton, Socket apis etc
- Ns3 library written in C++; Simulations are c++ executables; bindings in Python
- ns-3 uses the waf build system i.e., instead of `./configure;make`, type `./waf`
- Alignment with real systems (sockets, device driver interfaces)
- Alignment with input/output standards (pcap traces, ns-2 mobility scripts)
- Modular, documented core


Application
on
Sockets-like API
Protocol stack
Packet(s)
Node
NetDevice
ce
Channel
Channel
Application
on
Protocol stack
Node
NetDevice
ce

Simulation Basics

Simulation time moves discretely from event to event

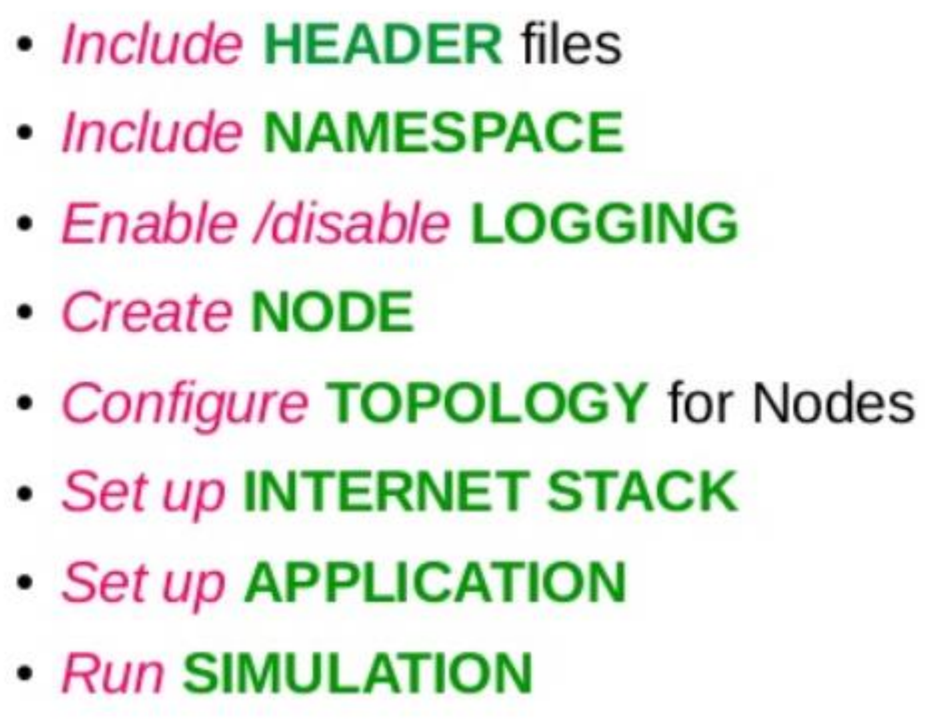
C++ functions schedule events to occur at specific simulation times

A simulation scheduler orders the event execution

Simulation::Run() gets it all started

Simulation stops at specific time or when events end

Main Program Structure



- *Include* **HEADER** files
- *Include* **NAMESPACE**
- *Enable /disable* **LOGGING**
- *Create* **NODE**
- *Configure* **TOPOLOGY** for Nodes
- *Set up* **INTERNET STACK**
- *Set up* **APPLICATION**
- *Run* **SIMULATION**

Code Walk Through